REMARKS

Claims 1, 4-21 and 31-40 are pending. Claims 1, 20, 21 and 31 are amended to more particularly point out the distinctions over the cited art. Claims 2, 3 are canceled. Claims 32-40 are withdrawn.

Election/Restriction

The Examiner has maintained the restriction requirement of the previous office action. Claims 32-40 have been withdrawn from consideration. To preserve the right of appeal, Applicants maintain traversal of the restriction requirement as stated in the previous office action.

Method allowing security policies to be defined in a distributed manner by components running in the system

Allowing security policies to be defined in a distributed manner by components running in the system, both to inform process partitioning decisions and to allow for application-specific security policies

Provide security and access control for objects and components using a capabilities model in conjunction with the use of object interfaces and enforcement of security using dynamic protection domains implemented using the process as a protection boundary around objects

Security model allows for each object in the system to declare its own security policies, and for the trust network to be implicitly and dynamically established by the application of each policy by each individual object

Principle of Least Authority

35 U.S.C. § 103 Rejection

Claims 1, 4-21 and 31 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Scheifler et al. (US Patent 6,138,238) in view of Colburn et al. (US Patent 6,173,404). Applicants respectfully traverse this rejection.

Scheifler – trusted source approach – source from a trusted developer

Determines access privileges based on principal and source of code

Applicants have Amended claims 1, 20, 21 and 31 to now recite "within the target object **and purely determined by the first interface**" (emphasis added). The Amendment shows how there is no checking of any information regarding the calling external object, rather the determination is purely based on the fact that the object was able to call the interface and based on the security policy associated with the first interface within the target object. This Amendment is fully supported by the following parts of the specification:

> *The decisions that each object makes as to how, when, and with whom to share its own interfaces or those of another object are up to the specific object implementation, and can be dependent on any number of factors that the object chooses to consider. In the preferred embodiment of the present invention, the* ***primary or sole factor used to determine which interfaces of a callee can be successfully requested by a caller is simply which interface the request was made using****. While other factors such as cryptographic signatures, password protection, and user verification may be used by any particular object to establish special rules based on particular situational needs, using these types of factors would require a central authority to be contacted and for certain tests to be performed on the caller to determine whether or not the caller has the necessary permissions.* ***If an object gets a call on one of its interfaces, the object knows that the calling object has authority or permission to make that call, by virtue of having had a reference to that interface in the first place****. In a preferred*

> *embodiment within any given object, the knowledge of* ***which particular interface was used to request another is all of the information that is needed to determine which interfaces get exposed in response to that request****, and thus how the trust network is built. Thus, constant permission checks with a central authority for each call to an object interface are not necessary.* (Paragraph 0052, emphasis added)
> And
>
> FIG. 6 is a flow diagram of a process for an object determining security access based on an interface call in accordance with a preferred embodiment of the present invention. At step 602 a target object receives a call at one of its interfaces from an external object. For example, the target interface may have three interfaces A, B, and C, each granting varying degrees of access to the target object's functions. Interface A granting the highest degree of access and interface C, the lowest. At step 604 the target object determines whether the external object is allowed access to other interfaces by checking its own security policies. In a preferred embodiment, the target object does not check with a central authority storing security data for all objects in the system. **If the external object is calling on interface A, the target object may determine that the external object has access to interfaces B and C. At step 606 the target object grants access to the external object to allow access to other interfaces as determined at step 604.** (Paragraph 0058, emphasis added)

Regarding claims 1, 20 – 21 and 31, Examine states that Scheifler teaches "a module configured with means for determining whether the external thread has access to other interfaces of the target object based on the call received at the first interface, wherein the determination is in association with implied permission;". However the Claim reads "the target object determining whether the external object has access to other interfaces of the target object based on the call to the first interface; wherein the determining step comprising examining a security policy contained entirely within the target object and purely determined by the first interface; and granting access to the other interfaces according to the determination." The Examiner relies on Scheifler Col 11 Line 20 through Col 13 Line 45, which relates to Protection domains. Though the present invention does contain protection domains, this claim does not relate to them and the protection domains as presented in the present invention differ from those in Scheifler.

The Examiner states that Scheifler doesn't teach determining other interfaces that a calling object has access to based on a security policy wholly located in the target object. As such the Examiner relies on Colburn. However Colburn like Scheifler depends on information that is greater than simply the call to a specific interface and the security policy associated with

that specific interface. This interpretation of Colburn and Scheifler is supported by, but not limited to the following references:

> *The owner identifier includes identification of the user, person, or entity (e.g., corporation) who or that creates the object, or identification of a computer system used by the user, person, or entity to create the object definition. The owner identifier provides a basis for distinguishing the creator of an object from the user of that object. This distinction allows instances of objects created by others to be given fewer access permissions or rights than the user implementing the object.* (Colburn Col 1 Lines 55 – 63)
> *and*

Conclusion

All of the stated grounds of rejection have been properly addressed. Applicants therefore respectfully request that the Examiner reconsider the outstanding rejections and allow the present claims. The Examiner is invited to telephone the undersigned representative if an interview might expedite allowance of this application.

Respectfully submitted,

BERRY& ASSOCIATES P.C.

Dated: July 26, 2010

By: /Shawn Diedtrich/
Shawn Diedtrich
Registration No. 58,176
Direct: 480.704.4615

9229 Sunset Blvd., Suite 630
Los Angeles, CA 90069
(310) 247-2860